PW SKILLS

Data Structure & Algorithms

Algorithm

↳ Sequence of finite steps

⇓ target

final outcome

in finite amount of time

Algorithm

Multiply two numbers

1) Take two num → a & b
2) Take c = a * b
3) Return & print c

Properties of Algorithms

↳ Terminate after finite amount of time

↳ Produce atleast one output

↳ Independent of any programming language

↳ unambiguous (Deterministic)

2 * 3
↳ 6

X
2 * 3
10

```
while (True):
    print('I am Priya')
```

Not a valid algorithm

↳ Infinite number of times

Data Structure

to store the data in a structured way

- ↳ Arrays
- ↳ Linked List
- ↳ Stack & Queue
- ↳ Tree & Graph

& many more

Time & Space complexity

Time Complexity → how many num of times any statement is running

↳ CPU Time

omega

1) Best case (Ω) — Big O
2) worst case (O)
3) Average case (Θ) — Theta

↳ Execute the code (Time)

↳ Lower Time Complexity

sum of natural num's. (n = 10)

Linear O(n)

(A) code 1

```
Sum = 0
for(i=0; i<n; i++)
{
    Sum += i;
}
return sum;
```

code 2 (B)

```
Sum = 0;
Sum = n*(n+1)/2
return sum;
```

constant O(1)

n = 10

```
for(i=0; i<n; i++){
    print('Hello')
}
```

print('Hello') → 10 times

↳ n num of times

$\frac{10*11}{2} = 55$

Example 1

O(n)

O(1) < O(n)

(B) — win >> (A)

$O(n^2)$

Quadratic Time complexity

n^2

Example 2

```
for(i=0; i<n; i++){
    for(j=0; j<n; j++)
    {
        Print('Hello')
    }
}
```

$n = 3$ — $n^2 = 3^2 = 9$

$i = 0$	$i = 1$	$i = 2$
$j = 0, 1, 2$	$j = 0, 1, 2$	$j = 0, 1, 2$
Hello — 3 times	Hello — 3 times	Hello - 3 times

Total = 9 times

Note:

1) Loops in a code
2) Bigger/Nested loops
3) Num of times (Iterations)
   (Statement is repeating)

Example 3

arr

0	1	2	3	4	5
10	20	30	40	50	60

arr(0) = 10

low = 0

high = 5

x = 20

Binary Search

Logarithmic Time Complexity

$O(\log_2 n)$

$mid = (0+5)/2 = 2$

arr(mid) = 30

Iteration	
n	0
$n/2$	1
$n/2^2$	2
⋮	⋮
k iteration	

$\frac{n}{2^k} = 1$ ——— k

↳ searched for (x)

$n = 2^k$

$\log_2 n = k \log_2 2$ (since $\log_2 2 = 1$)

$k = \log_2 n$